

Figure 1

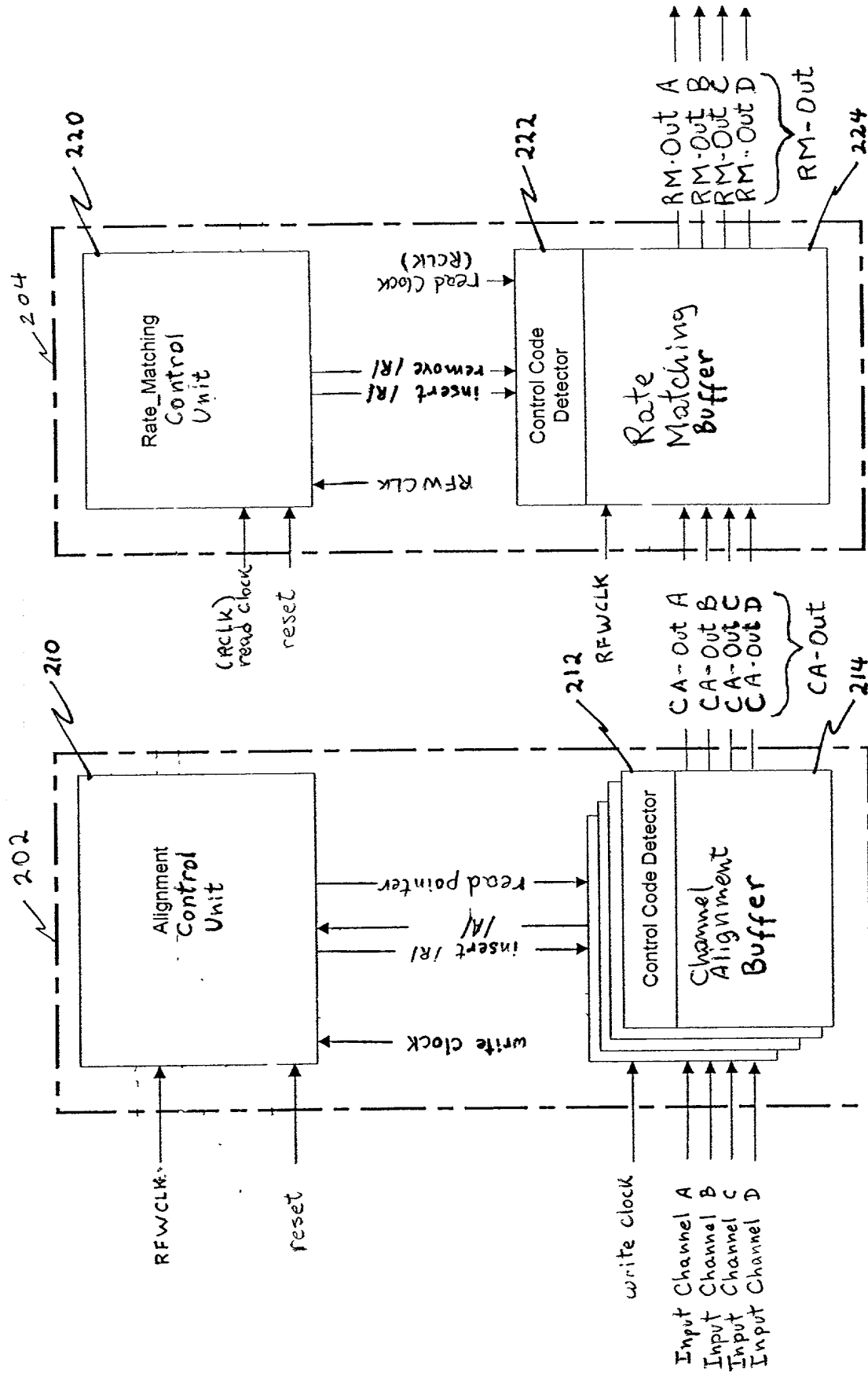


Figure 2

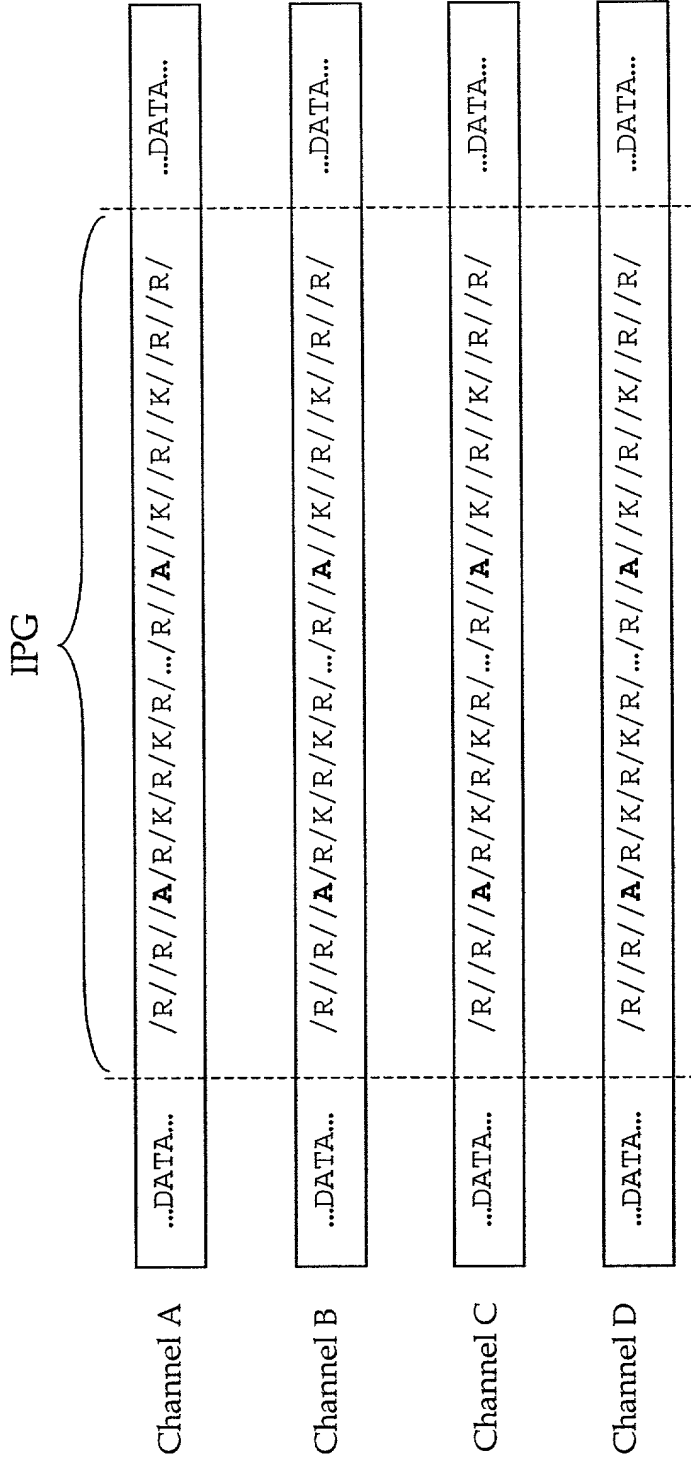


Figure 3

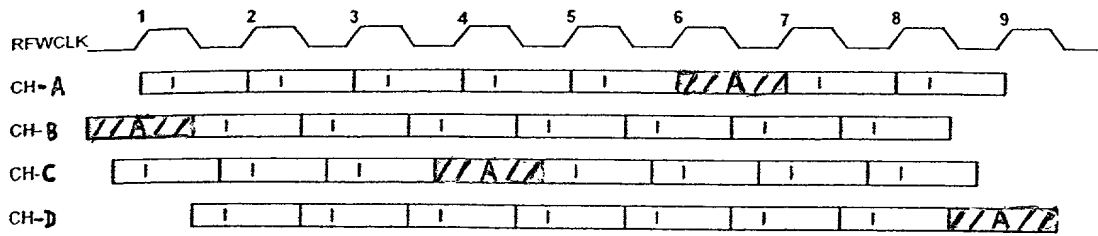


Figure 4A

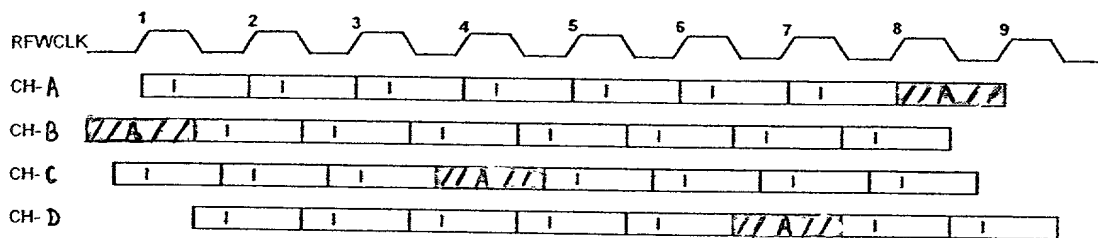


Figure 4B

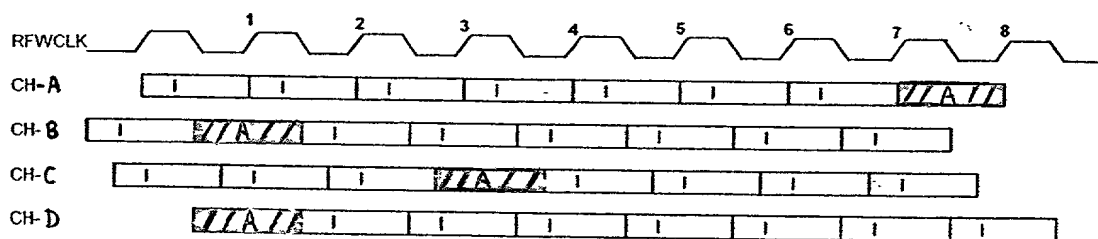


Figure 4C

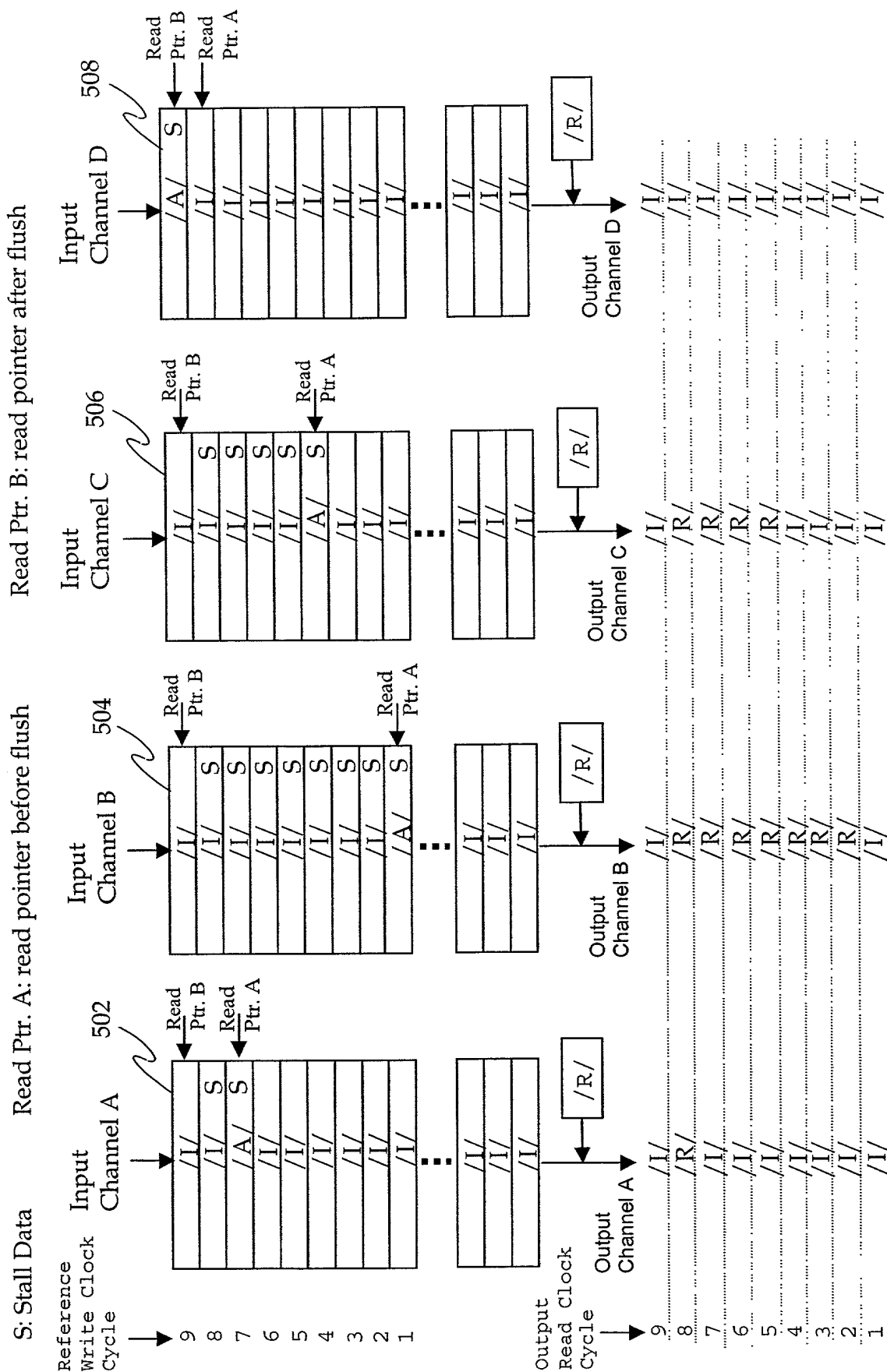


FIGURE 5

S: Stall Data Read Ptr.: read pointer

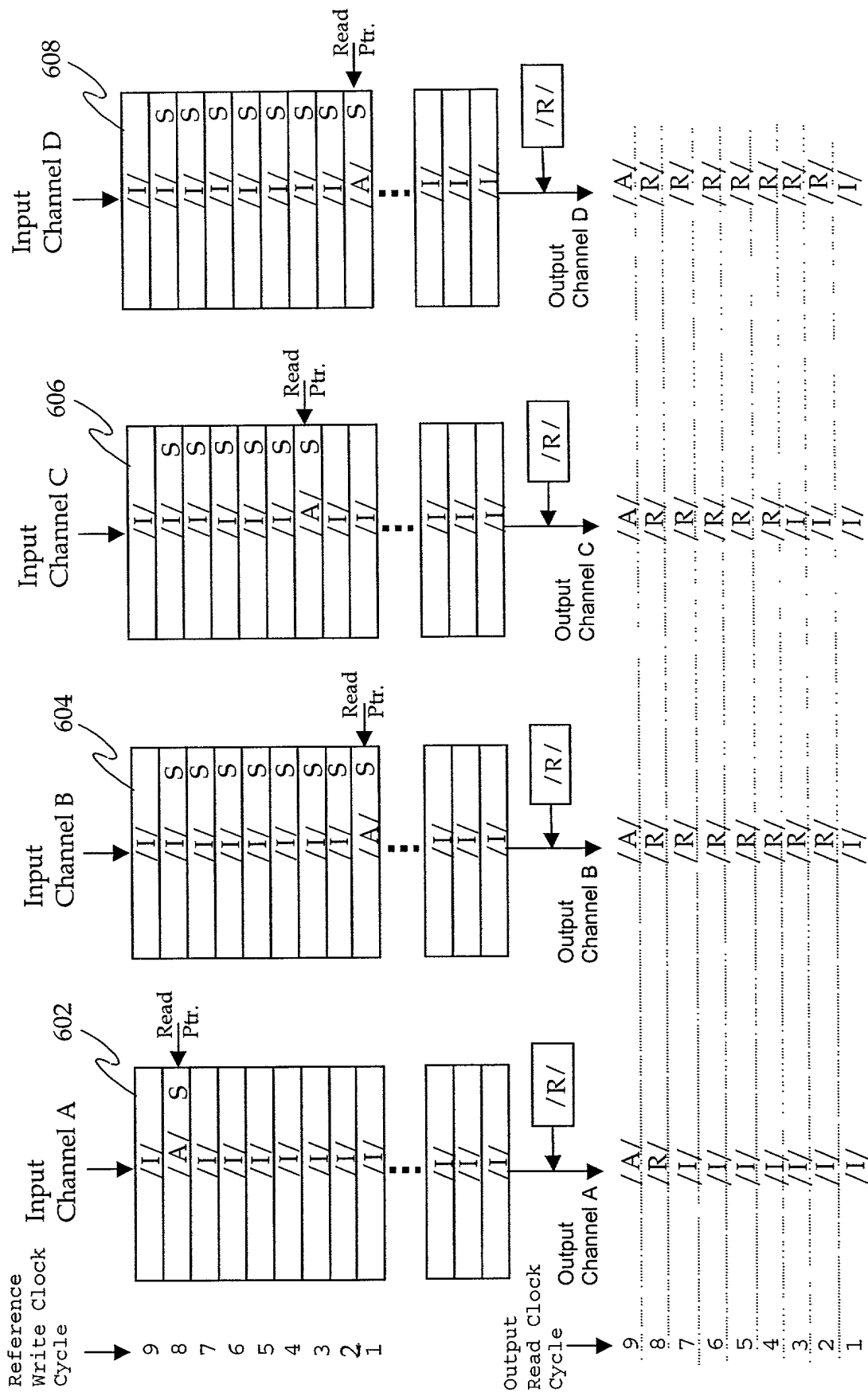


FIGURE 6

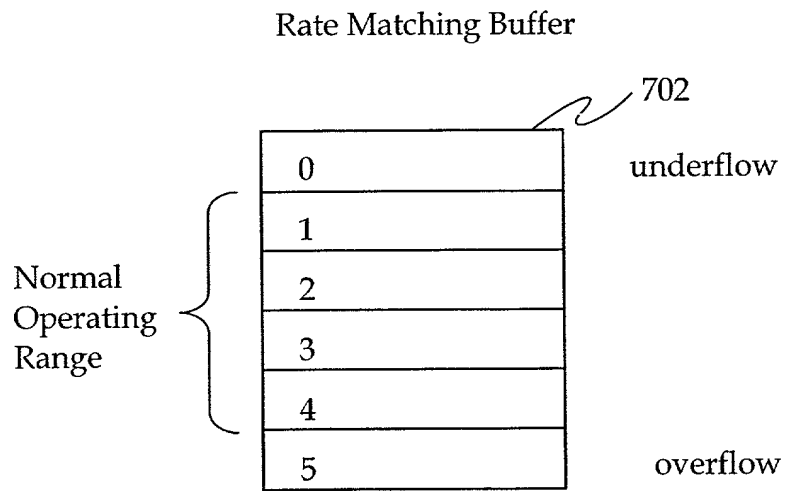


Figure 7A

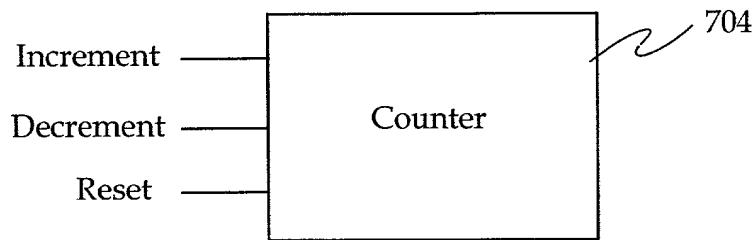


Figure 7B

Data in Buffer (max / min)	Buffer Data Count	Description
7 / 6	7	Overflow
6 / 5	6 (max. threshold)	Skip pending
5 / 4	5	Do nothing
4 / 3	4 (initial value)	Do nothing
3 / 2	3	Do nothing
2 / 1	2 (min. threshold)	Insert pending
1 / 0	1	Underflow

Figure 8

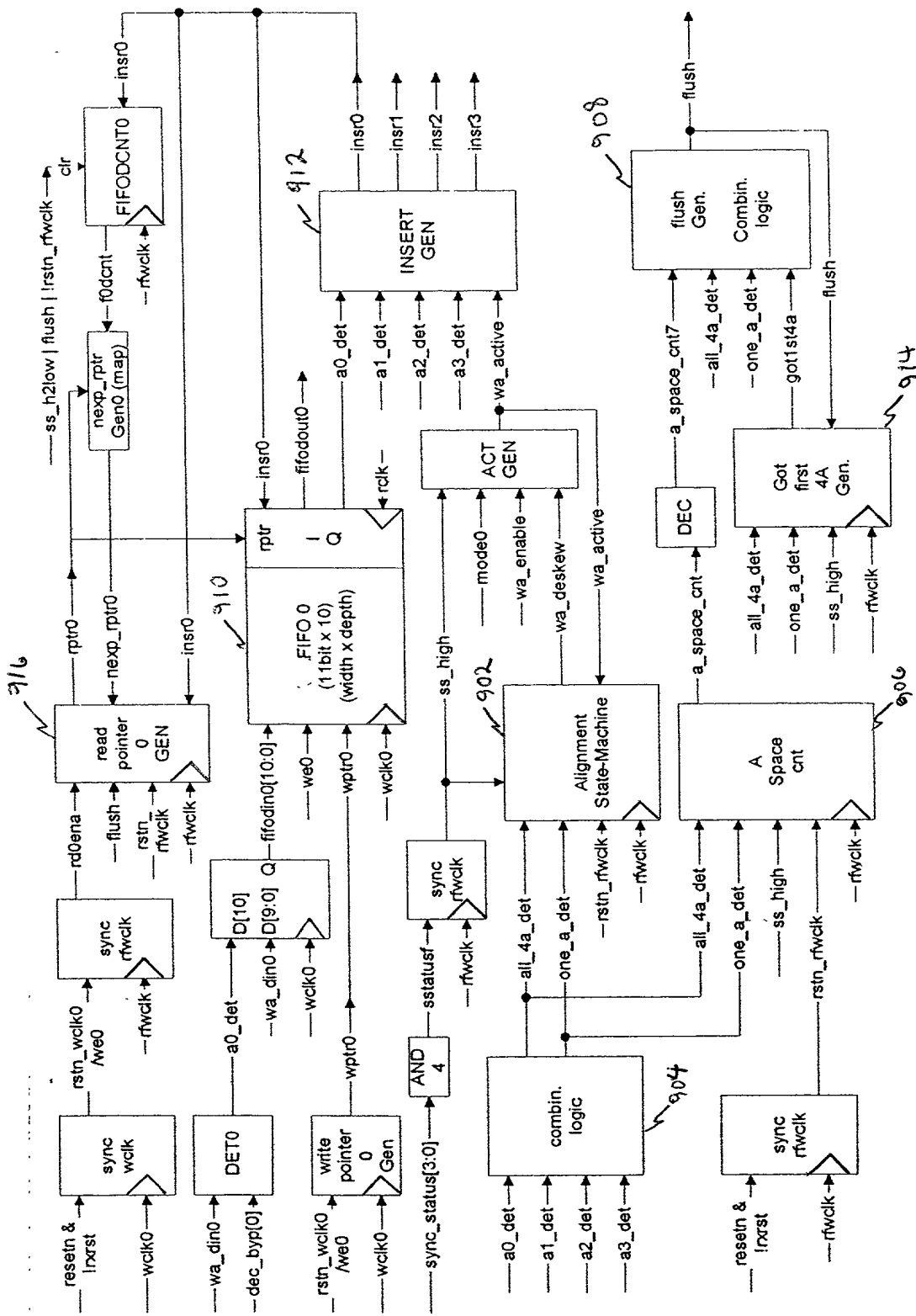


Figure 9A

Sub-Block Name	Description
Sync wclk	Synchronizer, to synchronize the reset to wclk domain. The synchronized reset is used as FIFO write enable.
Sync rfwclk	Synchronizer, to synchronize the reset to rfwclk domain.
Read pointer 0 GEN	Channel 0 read pointer. When insertion takes place, read pointer value will stay unchanged. When flush takes place, read pointer will be updated by the next expected pointer "nexp_rptr0".
Nexp rptr0 GEN	Next expected read pointer mapping table
FIFOzDCNT0	Channel 0 FIFO Data Counter
DET0	Channel 0 Align Character detector.
Write pointer 0 GEN	Channel 0 write pointer.
.FIFO 0	Alignment Channel 0 FIFO, a 10 x 11 - bit FIFO. IQ: The next FIFO output data.
AND4	Four input AND gate.
A Space cnt	Data skew counter. It counts from 0 to 7. When it equals to 7 and "A" characters ed, FIFO will be flushed.
ACT GEN	Word alignment active generator. A combinational logic.
INSERT GEN	Insert generation. A combinational logic.
Flush GEN	Flush generation. A combinational logic to flush stalled FIFO data. When data skew is greater than 8 columns, FIFO will be flushed to prevent from overflow.

Figure 9B

	Block Signal Interface	
Input Declaration		
input	wclk0;	//write clock for channel 0
input	wclk1;	//write clock for channel 1
input	wclk2;	//write clock for channel 2
input	wclk3;	//write clock for channel 3
input	rfwclk;	//reference write clock
input	resetn;	//asynchronous reset
input	wa-enable;	//word alignment enable
input	mode0;	//mode 0
input [3:0]	sync_status;	//synchronous status
input [3:0]	dec_byp;	//bypass decoder
input [9:0]	data_in0;	//channel 0 input data
input [9:0]	data_in1;	//channel 1 input data
input	data_in2;	//channel 2 input data
input [9:0]	data_in3;	//channel 3 input data
input	rxrstl	//receiver reset low assertion

Output Declaration		
output	wa_deskew;	//deskew status
output [9:0]	wa_out0;	//channel 0 output data
output [9:0]	wa_out1;	//channel 1 output data
output [9:0]	wa_out2;	//channel 2 output data
output [9:0]	wa_out3;	//channel 3 output data
output	wa_decby;	//decoder bypass
output	data_skew_err;	//data skew error

	FIFO Signal Interface	
Input declaration		
input	wclk;	//write clock
input [3:0]	wptr;	//write address 0-11
input [9:0]	din;	//incoming write data
input	we;	//write strobe, low assertion
input	rclk;	//read clock
input [3:0]	rptr;	//read address pointer
input	insr;	//insert R
input	decby;	//decoder bypass 1: bypass 0; enable

Output declaration		
output [9:0]	dout;	//read data
output	a_flag;	//indicate the next FIFO output data is an "A" character

	Next Reader Pointer Calculation Look-up Table	
input declaration		
input [3:0]	rptr;	//current read pointer
input [2:0]	fdcnt;	//current FIFO data count
output declaration		
output [3:0]	nexp_rptr;	//next expect read pointer

Figure 9C

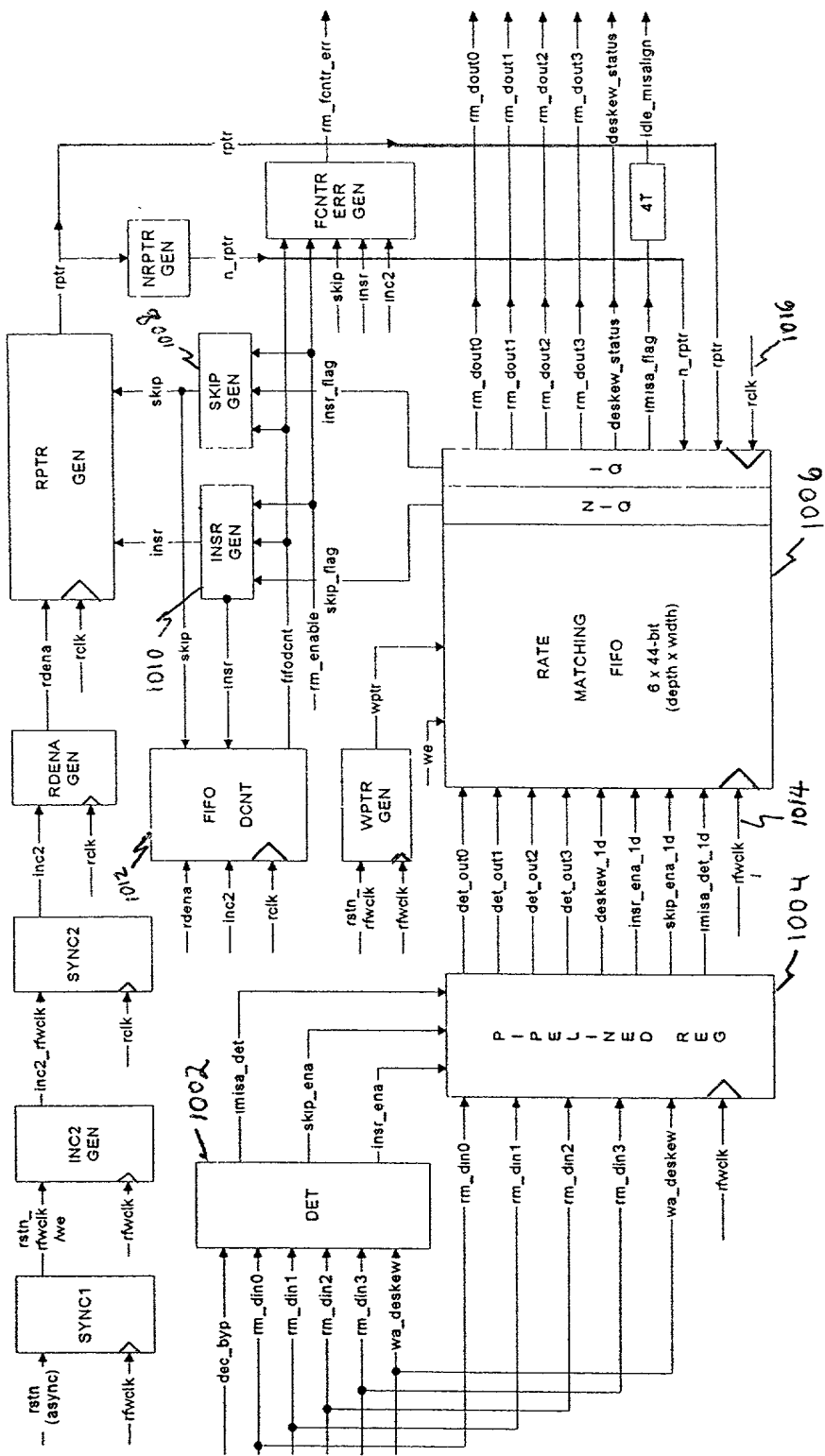


Figure 10A

Figure 8: Rate Matching RTL Block Diagram

Sub-Block Name	Description
SYNCl	Synchronizer, to synchronize the reset to "rfwclk" domain. The output of the synchronizer is the FIFO write enable.
INC2 GEN	Increment by-two generator. Therefore, write enable can be recognized by "rclk" side.
SYNC2	Synchronizer, to synchronize the increment by two "inc2_rfwclk" to rclk domain.
RDENA GEN	Read enable generator.
RPTR GEN	Read pointer generator. If insr = 1, read pointer stays unchanged. If skip = 1, read pointer will increment by two. Otherwise, increased by 1 on the clock edge.
DET	Detect IDLE characters and IDLE misalign at input stage. If 4 IDLEs are detected, insr_ena = 1. If wa_deskew = 1 and 4 R are detected or wa_deskew = 0 and 4 IDLEs are detected, skip_ena = 1. If detect IDLE misaligned, imisa_det = 1.
PIPELINED REG	The result of detector and data are registered (pipelined) prior written into the FIFO.
RMFIFO	Used to store data and the result of detection. Write data into the FIFO when write enable is asserted. Output data is addressed by the read pointer and clocked out on the next rising edge of rclk. If "insr" is asserted, 4 R will be presented at FIFO output regardless the value of the read pointer. IQ is the next data to be clocked out. 4 R can be inserted when bit-41 of IQ (insr_flag) is asserted. NIQ can be skipped when bit-42 of NIQ (skip_flag) is asserted.
4T	The width of IDLE misalign flag "imisa_flag" is extended to 4 rclk cycles wide for MDIO.
WPTR GEN	Write pointer generator
FIFO DCNT	FIFO Data CouNTer. The value of next FIFO data count is the sum of the current FIFO data count + (1 if insr else 0) + (2 if inc2 else 0) + (-1 if skip else 0) + (-1 if rdna else 0).
INSR GEN	Insertion generator, insr = 1 when all following condition are true: 1) rm_enable = 1 (rate matching enable is asserted). 2) fdcnt = 2 or fdcnt-ld = 2 (the current or previous FIFO data count equals to the minimum threshold (2)). 3) !insr-ld, not just inserted. (To avoid consecutive insertion) 4) unsr_flag = 1 (permission to insert, R will be inserted within IPG)
SKIP GEN	Deletion (skip) generator, skip = 1 when all the following condition are true: 1) rm_enable = 1 (rate matching enable is asserted). 2) fdcnt = 6 or fdcnt_ld = 6 (the current or previous FIFO data count equals to the maximum threshold (6)). 3) !skip_ld, not just skipped. (To avoid consecutive deletion). 4) skip flag = 1 (permission to skip, R column is available for skipping)
FCNTR ERR GEN	FIFO Data Counter Error Generator. To report FIFO overflow or underflow.
NRPTR GEN	Next read pointer generator.

Figure 10B

	Block Signal Interface	
Input declaration		
input	rfwclk;	//reference write clock
input	rclk;	//read clock
input	resetr;	//synchronous (rclk) reset
input	rm_enable;	//rate matching enable
input	dec_byp;	//by pass decoder
input [9:0]	rm_din0;	//rate matching channel 0 input data
input [9:0]	rm_din1;	//rate matching channel 1 input data
input	rm_din2;	//rate matching channel 2 input data
input	rm_din3;	//rate matching channel 3 input data
input	wa_deskew;	//deskew status
input	rxrst;	//receiver local reset

Output declaration		Description
output [9:0]	rm_dout0;	//rate matching channel 0 output data
output [9:0]	rm_dout1;	//rate matching channel 1 output data
output [9:0]	rm_dout2;	//rate matching channel 2 output data
output [9:0]	rm_dout3;	//rate matching channel 3 output data
output	idle_misalign;	//idle character misaligned
output	rm_fentr_err;	Rate matching FIFO data counter error
output	deskew_status;	//deskew status

FIFO Signal Interface		
Input declaration		
input	wclk;	//write clock
input	rclk;	//read clock
input	rstr;	//reset low assertion, sync to rclk
input [2:0]	wptr;	//write address 0 - 11
input [43:0]	din;	//incoming write data
input	we;	//write strobe, low assertion
input [2:0]	rptr;	//read address pointer
input [2:0]	n_rptr;	//next read address pointer
input	insr;	//insert R
input	wa_deskew;	//WA deskewed
input	decbyp;	//decoder bypass, used to insert R

Output declaration		
output [41:0]	dout;	//current FIFO output data, de-skew status, and idle misalign flag
output	insr_flag;	//insertion flag, to indicate the next FIFO output data are IDLE characters
output	skip_flag;	//skip flag, to indicate the next FIFO output data is delete-able

Figure 10C